

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



(BLM4531) Ağ Tabanlı Teknolojiler Ve Uygulamaları Dersi I

Uygulama Dökümanı

Şehmus Altaş

20291316

14/01/2026

Github: [GitHub - SehmusAlts/recipe_app_.net](https://github.com/SehmusAlts/recipe_app_.net)

Video Link:

<https://drive.google.com/drive/folders/1H9ETdqYVznB0tnQXfOvdd696TdIZW8b9?usp=sharing>

1. GİRİŞ VE PROJE ÖZETİ

Bu proje kapsamında ASP.NET Core (.NET 9.0) kullanılarak geliştirilen RecipeApp, kullanıcıların yemek tariflerini görüntüleyebildiği, kendi tariflerini ekleyebildiği, tarifleri favorilere alıp puanlayabildiği tam kapsamlı bir web uygulamasıdır. Proje, hem RESTful API hem de MVC tabanlı web arayüzü içerecek şekilde full-stack olarak tasarlanmıştır.

Uygulamada kullanıcılar e-posta ve şifre bilgileri ile kayıt olup giriş yapabilmekte, sisteme giriş yaptıktan sonra tarif oluşturma, düzenleme ve silme işlemlerini gerçekleştirebilmektedir. Kullanıcılar ayrıca diğer tarifleri inceleyebilmekte, tariflere puan verebilmekte ve favori listelerini yönetebilmektedir. Harici bir veri kaynağı olarak DummyJSON API kullanılarak tarif senkronizasyonu da desteklenmektedir.

Proje, Clean Architecture (Onion Architecture) yaklaşımı ile geliştirilmiş olup iş mantığı, veri erişimi ve sunum katmanları birbirinden net bir şekilde ayrılmıştır. Veritabanı yapısı Entity Framework Core Code-First yaklaşımı ile oluşturulmakta ve uygulama ilk çalıştırıldığında otomatik olarak veritabanı şeması üretilmektedir. Bu dokümantasyon, projenin mimari yapısını, kullanılan teknolojileri ve temel teknik kararlarını açıklamayı amaçlamaktadır.

2. TEKNOLOJİLER VE FRAMEWORK'LER

RecipeApp projesi, modern .NET ekosistemine uygun, güncel ve sürdürülebilir teknolojiler kullanılarak geliştirilmiştir. Backend tarafında .NET 9.0 framework'ü tercih edilmiş, uygulamanın servis yapısı ASP.NET Core altyapısı üzerine inşa edilmiştir. RESTful mimariyi desteklemek amacıyla ASP.NET Core Web API kullanılmış, kullanıcıya sunulan web arayüzü ise ASP.NET Core MVC yapısı ile geliştirilmiştir. Bu yaklaşım sayesinde API ve web katmanları birbirinden ayrılmış, farklı istemcilerin aynı API'yi kullanabilmesi mümkün hale getirilmiştir.

Veri yönetimi için SQL Server tabanlı ilişkisel bir veritabanı kullanılmıştır. Nesne-ilişkisel eşleme işlemleri Entity Framework Core aracılığıyla gerçekleştirilmiştir. Veritabanı tasarımı Code-First yaklaşımı ile yapılmış olup, entity sınıfları üzerinden veritabanı şeması otomatik olarak oluşturulmaktadır. Migration mekanizması sayesinde veritabanı şemasındaki değişiklikler kontrollü bir şekilde yönetilmekte ve versiyonlama sağlanmaktadır. Uygulama

çalıştığı anda gerekli migration'lar uygulanarak veritabanı otomatik olarak hazır hale getirilmektedir.

Kimlik doğrulama ve yetkilendirme işlemleri için JWT tabanlı authentication mekanizması uygulanmıştır. Kullanıcı giriş işlemleri sonucunda üretilen token'lar, istemci tarafında saklanmakta ve API isteklerinde kullanılarak güvenli erişim sağlanmaktadır. Kullanıcı parolaları, güvenlik amacıyla BCrypt algoritması kullanılarak hash'lenmekte ve düz metin olarak saklanmamaktadır. Bu sayede parola güvenliği üst seviyede tutulmaktadır.

Uygulamaya gelen verilerin doğruluğunu sağlamak amacıyla FluentValidation kütüphanesi kullanılmıştır. Kullanıcıdan alınan tüm giriş verileri, ilgili validator sınıfları aracılığıyla kontrol edilmektedir. Nesneler arası dönüşümler için AutoMapper kullanılmış, bu sayede entity ve DTO yapıları arasındaki dönüşümler merkezi ve düzenli bir şekilde yönetilmiştir.

Frontend tarafında ASP.NET Core MVC ile birlikte Razor View Engine kullanılmıştır. Kullanıcı arayüzü Bootstrap 5 ile oluşturulmuş, responsive tasarım prensipleri uygulanarak farklı ekran boyutlarına uyum sağlanmıştır. Temel istemci tarafı etkileşimleri için jQuery kullanılmıştır. API uç noktalarının test edilebilmesi ve dokümantasyonu için Swagger (OpenAPI) entegrasyonu sağlanmıştır.

Frontend tarafında ASP.NET Core MVC, Razor View Engine ve Bootstrap 5 kullanılmıştır. Responsive tasarım ve temel etkileşimler Bootstrap ve jQuery ile sağlanmıştır. API dokümantasyonu için Swagger (OpenAPI) entegre edilmiştir.

3. MİMARİ TASARIM

RecipeApp projesi, Clean Architecture (Onion Architecture) prensipleri temel alınarak tasarlanmıştır. Bu mimari yaklaşım sayesinde uygulamanın iş kuralları, altyapı ve sunum katmanlarından bağımsız hale getirilmiştir. Katmanlar arası bağımlılıklar içten dışa doğru olacak şekilde düzenlenmiş, merkezde yer alan domain katmanı hiçbir dış bağımlılık içermeyecek şekilde yapılandırılmıştır. Bu sayede iş mantığı, kullanılan framework veya veritabanı teknolojilerinden etkilenmeden çalışabilmektedir.

Uygulamanın iş mantığı servis katmanı üzerinden yönetilmektedir. Controller sınıfları yalnızca HTTP isteklerini karşılamakta, gelen istekleri ilgili servislere yönlendirmekte ve sonuçları istemciye döndürmektedir. İş kurallarının controller'lar içerisinde yer almaması, kodun daha okunabilir ve sürdürülebilir olmasını sağlamaktadır. Veri erişim işlemleri repository yapıları üzerinden gerçekleştirilmekte, Unit of Work yaklaşımı ile birden fazla veri işlemi tek bir işlem bütünlüğü altında yönetilmektedir.

Bu mimari yapı, test edilebilirliği önemli ölçüde artırmaktadır. Servisler ve repository'ler interface'ler üzerinden çalıştığı için, birim testlerde gerçek veritabanı veya altyapı bağımlılıkları olmadan mock yapılar kullanılabilir. Ayrıca bu yaklaşım, ileride farklı bir veritabanı ya da harici servis kullanılması durumunda minimum kod değişikliği ile sistemin güncellenmesine olanak tanımaktadır.

Bağımlılıkların yönetimi için ASP.NET Core'un sunduğu Dependency Injection mekanizması kullanılmıştır. Tüm servisler, repository'ler ve altyapı bileşenleri constructor injection yöntemi ile sisteme dahil edilmiştir. Bu yapı sayesinde sınıflar arasında sıkı bağımlılıkların önüne geçilmiş ve loosely coupled bir mimari elde edilmiştir. Sonuç olarak proje, ölçeklenebilir, bakımı kolay ve uzun vadede geliştirilebilir bir yapıya sahip hale getirilmiştir.

4. VERİTABANI TASARIM

RecipeApp projesinde veri kalıcılığı için SQL Server tabanlı ilişkisel bir veritabanı kullanılmıştır. Veritabanı tasarımı, Entity Framework Core aracılığıyla Code-First yaklaşımı ile gerçekleştirilmiştir. Bu yaklaşım sayesinde veritabanı şeması, uygulama içerisindeki entity sınıfları üzerinden otomatik olarak oluşturulmakta ve uygulama çalıştığı anda migration mekanizması ile güncel hale getirilmektedir. Böylece manuel veritabanı oluşturma ihtiyacı ortadan kaldırılmıştır.

Veritabanı yapısı, kullanıcılar, tarifler, puanlar ve favoriler etrafında kurgulanmıştır. Kullanıcı bilgileri User entity'si üzerinden tutulmakta, tariflere ait tüm temel bilgiler Recipe entity'si içerisinde saklanmaktadır. Tarifler ile kullanıcılar arasında bir ilişki kurulmuş olup, bir kullanıcının birden fazla tarif oluşturabilmesi desteklenmektedir. Puanlama ve favori işlemleri ise ayrı entity'ler üzerinden yönetilerek çoktan çoğa ilişkiler kontrollü bir şekilde modellenmiştir.

Uygulamada harici API üzerinden alınan tarifler ile kullanıcı tarafından oluşturulan tarifler birbirinden ayrıştırılmıştır. Harici API'den gelen tarifler özel alanlar ile işaretlenmekte ve bu tariflerin kullanıcılar tarafından düzenlenmesi veya silinmesi engellenmektedir. Bu yapı, sistem bütünlüğünü korumak ve harici veriler üzerinde kontrolsüz değişikliklerin önüne geçmek amacıyla tercih edilmiştir.

Veri silme işlemlerinde soft delete yaklaşımı uygulanmıştır. Kayıtlar fiziksel olarak veritabanından silinmemekte, bunun yerine ilgili kayıtlar pasif hale getirilmektedir. Bu sayede veri kaybı önlenmiş, gerektiğinde silinen kayıtların tekrar erişilebilir olması

sağlanmıştır. Global query filter'lar kullanılarak pasif kayıtlar varsayılan sorguların dışında tutulmaktadır.

Veritabanı performansını artırmak amacıyla sık kullanılan alanlar üzerinde index yapıları oluşturulmuştur. Özellikle kullanıcı e-posta bilgileri, tarif isimleri ve kategori alanları üzerinde indeksleme yapılmıştır. Bu sayede listeleme, filtreleme ve arama işlemleri daha verimli hale getirilmiştir.

5. UYGULAMA ÖZELLİKLERİ VE FONKSİYONLAR

RecipeApp, kullanıcıların tarif içerikleri ile etkileşime girebildiği kapsamlı bir web uygulaması olarak tasarlanmıştır. Uygulama içerisinde kullanıcılar sisteme kayıt olabilmekte ve e-posta ile şifre bilgileri üzerinden giriş yapabilmektedir. Kimlik doğrulama işlemleri JWT tabanlı olarak yürütülmekte, giriş yapan kullanıcılar için yetkilendirme kontrolleri uygulanmaktadır. Oturum açmayan kullanıcılar yalnızca tarifleri görüntüleyebilmekte, içerik ekleme veya etkileşim özelliklerine erişememektedir.

Tarif yönetimi uygulamanın temel işlevlerinden biridir. Kullanıcılar sisteme giriş yaptıktan sonra kendi tariflerini oluşturabilmekte, mevcut tariflerini güncelleyebilmekte ve silebilmektedir. Bu işlemler sırasında yalnızca tarifi oluşturan kullanıcının ilgili tarif üzerinde işlem yapmasına izin verilmektedir. Tarifler; açıklama, malzeme listesi, hazırlanış adımları, kategori bilgisi ve görsel gibi alanları içermektedir. Listeleme işlemlerinde sayfalama desteği sunularak performanslı veri erişimi sağlanmaktadır.

Uygulama içerisinde favori yönetimi özelliği bulunmaktadır. Kullanıcılar beğendikleri tarifleri favorilerine ekleyebilmekte ve istedikleri zaman favorilerinden çıkarabilmektedir. Favoriye alınan tarifler kullanıcıya özel olarak saklanmakta ve ayrı bir ekranda listelenmektedir. Aynı tarifi bir kullanıcı tarafından birden fazla kez favoriye eklenmesi sistem tarafından engellenmektedir.

Tariflere puan verme ve deęerlendirme özellięi de uygulamanın önemli işlevlerinden biridir. Kullanıcılar tariflere belirlenen puan aralığında deęerlendirme yapabilmektedir. Her kullanıcı bir tarife yalnızca bir kez puan verebilmekte, daha sonra bu puanı güncelleyebilmektedir. Tariflerin ortalama puan bilgileri sistem tarafından hesaplanmakta ve kullanıcı arayüzünde gösterilmektedir.

Uygulama, harici bir veri kaynaęı ile entegre çalışabilmektedir. DummyJSON API üzerinden alınan tarifler sisteme senkronize edilmekte ve yerel tariflerden ayrıştırılmaktadır. Harici kaynaktan gelen tarifler yalnızca görüntülenebilir nitelikte olup kullanıcılar tarafından düzenlenememekte veya silinememektedir. Bu sayede hem hazır içerik sağlanmakta hem de kullanıcı tarafından oluşturulan içerikler ile sistem bütünlüğü korunmaktadır.

6. GÜVENLİK VE PERFORMANS

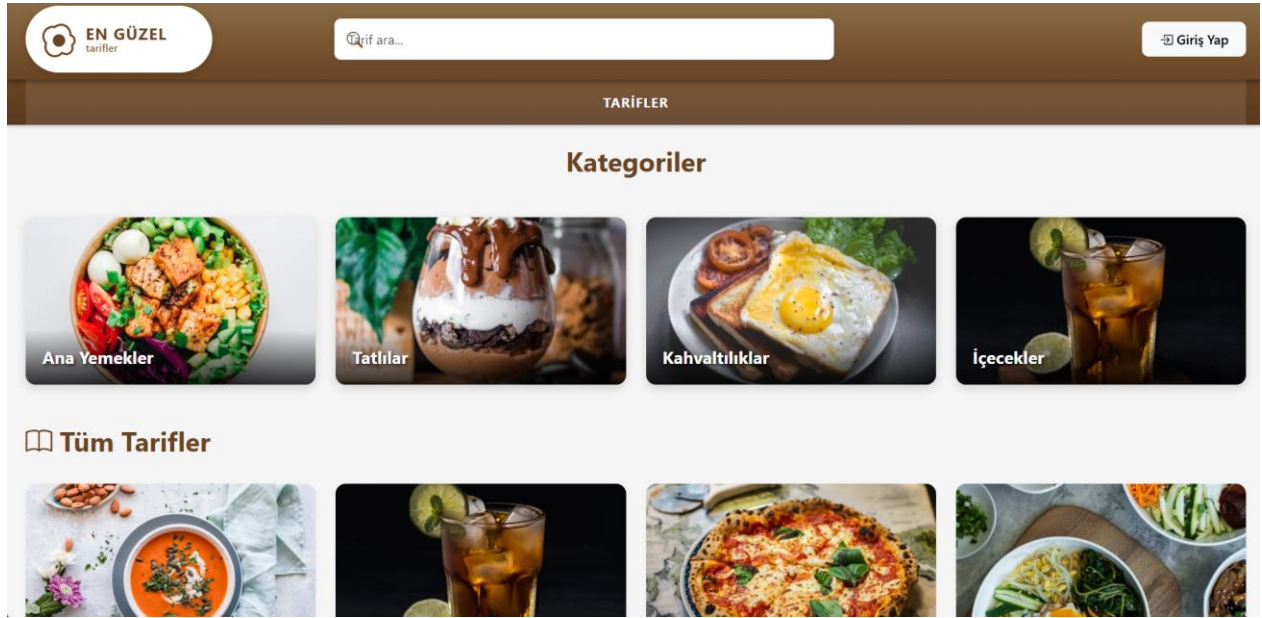
RecipeApp projesinde güvenlik, uygulamanın temel tasarım kriterlerinden biri olarak ele alınmıştır. Kimlik doğrulama işlemleri JWT tabanlı authentication mekanizması ile sağlanmaktadır. Kullanıcı giriş işlemi sonrasında üretilen token, istemci tarafında saklanmakta ve yetkilendirme gerektiren tüm API isteklerinde kullanılmaktadır. Bu yapı sayesinde yetkisiz erişimlerin önüne geçilmektedir.

Kullanıcı parolaları güvenli bir şekilde saklanabilmesi amacıyla BCrypt algoritması ile hash'lenmektedir. BCrypt, her parola için benzersiz salt kullanarak brute-force saldırılarına karşı güçlü bir koruma sağlamaktadır. Parolalar hiçbir aşamada düz metin olarak tutulmamaktadır. API endpoint'leri, yetkilendirme gereksinimlerine göre koruma altına alınmış ve yalnızca yetkili kullanıcıların belirli işlemleri gerçekleştirmesine izin verilmiştir.

Giriş doğrulama ve veri güvenliği açısından input validation önemli bir rol oynamaktadır. Uygulamaya gelen tüm kullanıcı girdileri FluentValidation kullanılarak kontrol edilmekte, geçersiz veya eksik verilerin sisteme girmesi engellenmektedir. Bu yaklaşım, hem güvenlik açıklarını azaltmakta hem de uygulama kararlılığını artırmaktadır.

Performans tarafında, veritabanı işlemlerinin tamamı asenkron olarak gerçekleştirilmiştir. async/await yapısı kullanılarak I/O işlemlerinde thread bloklanmasının önüne geçilmiştir. Listeleme işlemlerinde sayfalama mekanizması uygulanmış, büyük veri setlerinin tek seferde yüklenmesi engellenmiştir. Sık kullanılan alanlar üzerinde oluşturulan index'ler sayesinde sorgu performansı artırılmıştır.

Soft delete yaklaşımı ve global query filter kullanımı, hem veri güvenliğini hem de performansı olumlu yönde etkilemektedir. Silinen kayıtlar fiziksel olarak kaldırılmadığı için veri kaybı riski azaltılmış, aynı zamanda varsayılan sorgularda bu kayıtlar otomatik olarak filtrelenerek performans kaybı önlenmiştir.



EN GÜZEL
tarifler

Tarif ara...

Giriş Yap

TARİFLER

Giriş Yap

Email

ornek@email.com

Password

☐ Beni Hatırla


Giriş Yap

Hesabınız yok mu? [Kayıt olun](#)

EN GÜZEL
tarifler

Tarif ara...

Giriş Yap



Çikolatalı Kurabiye

☆☆☆☆☆ 0,0 (0 değerlendirme)

Tatlı

Yumuşacık damla çikolatalı kurabiye.

Hazırlık
15 dk

Pişirme
15 dk

Porsiyon
20 kişilik

Malzemeler

✓ 2 su bardağı un

✓ 125 gr tereyağı

✓ 1 su bardağı şeker

Hazırlanışı

Tereyağı ve şekerini çırpın. Yumurta ekleyin. Kuru malzemeleri karıştırıp ekleyin. Şekil verip 180 derecede 15 dakika pişirin.